Delivering With Project Builder

Session 908

Delivering With Project Builder

Anders Bertelrud
Tech Lead, Project Builder

Introduction

So, you have a project that builds . . . now what?

Introduction

So, you have a project that builds . . . now what?

- Streamline the build process

Introduction

So, you have a project that builds . . now what?

- Streamline the build process
- Polish your product

Introduction

So, you have a project that builds . . now what?

- Streamline the build process
- Polish your product
- Build for deployment

Introduction

So, you have a project that builds . . now what?

- Streamline the build process
- Polish your product
- Build for deployment
- Package your product

Introduction

So, you have a project that builds . . now what?

- Streamline the build process
- Polish your product
- Build for deployment
- Package your product
- Ship it!

What You Will Learn

- Using the IDE effectively
- Speeding up GCC 3 compilation
- Adding the finishing touches to your product
- Building your product for deployment
- Creating disk images and Installer packages

What You Will Learn

- Using the IDE effectively
- Speeding up GCC 3 compilation
- Adding the finishing touches to your product
- Building your product for deployment
- Creating disk images and Installer packages

Using the IDE Effectively

- Customize your builds using *build styles*
- Set up a shared product directory
    - Access another project's products
    - Keep binaries separate from source code
- Create multiple execution contexts
    - Quickly switch between test cases

Build Styles

- Build targets in different ways
- Examples:
    - Debugging symbols on or off
    - Assertions enabled or disabled
    - Different optimization levels
    - Alternate algorithms
    - Features enabled or disabled

What Do Build Styles Do?

- Dynamically modify target settings at build time
- Each setting in a build style can either:
  - Override the target's setting
    - **DEBUGGING_SYMBOLS = NO**
  - Append to the target's setting
    - **OTHER_CFLAGS += -DUSE_HASH_TABLES**
- Target can also refer back to build style's settings
- But: circular references are not permitted

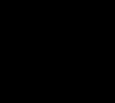
Precedence of Build Settings

Using Build Styles

- The IDE always has one active build style
- The pbxbuild tool allows greater flexibility:
  - **pbxbuild install**
  - **pbxbuild install -buildstyle “Deployment”**
  - **pbxbuild -buildstyle “EnableExtraAsserts”**
  - **pbxbuild -activebuildstyle**

Shared Product Directory

- Why?
  - Allows a project to access products built by a different project
  - Speed: source code on network, binaries local



Shared Product Directory

- Why?
  - Allows a project to access products built by a different project
  - Speed: source code on network, binaries local
- How?
  - Specify a directory for built products
  - All generated binaries will be put there
  - Runtime environment is set up appropriately

What You Will Learn

- Using the IDE effectively
- Speeding up GCC 3 compilation
- Adding the finishing touches to your product
- Building your product for deployment
- Creating disk images and Installer packages

What You Will Learn

- Using the IDE effectively
- Speeding up GCC 3 compilation
- Adding the finishing touches to your product
- Building your product for deployment
- Creating disk images and Installer packages

Speeding Up GCC 3 Compiles

- Step 1: select GCC 3 as your compiler
  - Set **USE_GCC3=YES**

Speeding Up GCC 3 Compiles

- Step 1: select GCC 3 as your compiler
  - Set **USE_GCC3=YES**
- Step 2: define a prefix file
  - Also specify that it should be precompiled

Speeding Up GCC 3 Compiles

- Step 1: select GCC 3 as your compiler
  - Set **USE_GCC3=YES**
- Step 2: define a prefix file
  - Also specify that it should be precompiled
- Step 3: enable PFE support
  - **USE_GCC3_PFE_SUPPORT=YES**

Speeding Up GCC 3 Compiles

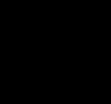
- Step 1: select GCC 3 as your compiler
  - Set **USE_GCC3=YES**
- Step 2: define a prefix file
  - Also specify that it should be precompiled
- Step 3: enable PFE support
  - **USE_GCC3_PFE_SUPPORT=YES**
- But: GCC 3 is still a prerelease
  - Don’t ship products built with it until Jaguar

What You Will Learn

- Using the IDE effectively
- Speeding up GCC 3 compilation
- Adding the finishing touches to your product
- Building your product for deployment
- Creating disk images and Installer packages

What You Will Learn

- Using the IDE effectively
- Speeding up GCC 3 compilation
- Adding the finishing touches to your product
- Building your product for deployment
- Creating disk images and Installer packages



Adding Finishing Touches

- Versioning
  - Built-in support for Apple Generic Versioning
  - Optional integration with CVS

Adding Finishing Touches

- Versioning
  - Built-in support for Apple Generic Versioning
  - Optional integration with CVS
- Regenerating localizable strings
  - Use a “Legacy Target” to run *genstrings*

Adding Finishing Touches

- Versioning
  - Built-in support for Apple Generic Versioning
  - Optional integration with CVS
- Regenerating localizable strings
  - Use a “Legacy Target” to run *genstrings*
- Scatterloading
  - Code order is based on runtime use patterns

Embedding Frameworks

- Why?
  - Simpler packaging (better user experience)
  - Robustness (frameworks aren't left behind)

Embedding Frameworks

- Why?
  - Simpler packaging (better user experience)
  - Robustness (frameworks aren't left behind)
- What are the alternatives?
  - Shared source code
  - Static library

Embedding Frameworks

- Why not just a static library?
  - Exposure of plug-in API
  - Development granularity
  - Packaging of resources and headers

Embedding Frameworks

- Setting up the framework
  - Special installation location
    - **@executable_path/../Frameworks**
  - Set **SKIP_INSTALL=YES**

Embedding Frameworks

- Setting up the framework
  - Special installation location
    - **@executable_path/../Frameworks**
  - Set **SKIP_INSTALL=YES**
- Setting up the application
  - Copy-Files build phase to embed framework

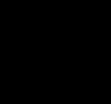
Demo

Deploying and Packaging

Christian Molick

What You Will Learn

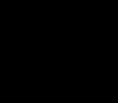
- Using the IDE effectively
- Speeding up GCC 3 compilation
- Adding the finishing touches to your product
- Building your product for deployment
- Creating disk images and Installer packages



What You Will Learn

- Using the IDE effectively
- Speeding up GCC 3 compilation
- Adding the finishing touches to your product
- Building your product for deployment
- Creating disk images and Installer packages

Deployment Builds



Deployment Builds

- Compile optimization and macros

Deployment Builds

- Compile optimization and macros
- Symbol editing and stripping

Deployment Builds

- Compile optimization and macros
- Symbol editing and stripping
- Avoiding non-deployment files

Deployment Builds

- Compile optimization and macros
- Symbol editing and stripping
- Avoiding non-deployment files
- Setting ownership and permissions

Deployment Builds

- Compile optimization and macros
- Symbol editing and stripping
- Avoiding non-deployment files
- Setting ownership and permissions
- Distribution folder layout

Compile Optimization & Macros

- Compile optimization
  - Optimization can interfere with debugging
  - Recommended default is **-Os**
- Macros
  - **OTHER_CFLAGS=-D_NO_CRASH_RISK**

Symbol Editing and Stripping

- Edit exported symbols
- Strip symbols from products
- Strip embedded content

Avoiding Non-Deployment Files

- Non-deployment files
  - Intermediates
  - Development property lists
- Recommended procedures
  - Clean before building for deployment
  - Check deployment build results

File Ownership and Permissions

- Owner and group settings
  - System content: root/wheel
  - Other content: root/admin
- File mode settings
  - Readable and executable by everyone
  - Writable by owner only

Distribution Folder Layout

- Deployment builds may have a structure
- Distribution roots may represent the system root

Build Actions

Build Actions

- Determine default values for build settings

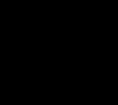
Build Actions

- Determine default values for build settings
- Determine the build process

Build Actions

- Determine default values for build settings
- Determine the build process
- Actions **clean** and **installsrc** are special cases

Build Styles for Deployment



Build Styles for Deployment

- Build styles determine how targets are built

Build Styles for Deployment

- Build styles determine how targets are built
- Build style settings override target settings

Demo

Deploying and Packaging

Anders Bertlrud

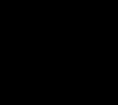
What You Will Learn

- Using the IDE effectively
- Speeding up GCC 3 compilation
- Adding the finishing touches to your product
- Building your product for deployment
- Creating disk images and Installer packages

What You Will Learn

- Using the IDE effectively
- Speeding up GCC 3 compilation
- Adding the finishing touches to your product
- Building your product for deployment
- Creating disk images and Installer packages

Packaging for Distribution



Packaging for Distribution

- Disk images

Packaging for Distribution

- Disk images
    - Compact and accessible

Packaging for Distribution

- Disk images
  - Compact and accessible
  - Can do simple installation processing

Packaging for Distribution

- Disk images
  - Compact and accessible
  - Can do simple installation processing
- Installer packages for special cases

Packaging for Distribution

- Disk images
  - Compact and accessible
  - Can do simple installation processing
- Installer packages for special cases
  - Scattered files

Packaging for Distribution

- Disk images
  - Compact and accessible
  - Can do simple installation processing
- Installer packages for special cases
  - Scattered files
  - Installation processing

Making Disk Images

Making Disk Images

- Create with **hdiutil**

Making Disk Images

- Create with **hdiutil**
- Make file system

Making Disk Images

- Create with **hdiutil**
- Make file system
- Mount with **hdid**

Making Disk Images

- Create with **hdiutil**
- Make file system
- Mount with **hdid**
- Populate using any copy commands

Making Disk Images

- Create with **hdiutil**
- Make file system
- Mount with **hdid**
- Populate using any copy commands
- Compress with **hdiutil**

Automating Packaging

- Alternate target
- Script phase

Demo

In Conclusion

Anders Bertlrud

Summary

We have . . .

Summary

We have . . .

- Streamlined the build process

Summary

We have . . .

- Streamlined the build process
- Polished the product

Summary

We have . . .

- Streamlined the build process
- Polished the product
- Built for deployment

Summary

We have . . .

- Streamlined the build process
- Polished the product
- Built for deployment
- Packaged the product

Summary

We have . . .

- Streamlined the build process
- Polished the product
- Built for deployment
- Packaged the product

Now . . .

Summary

We have . . .

- Streamlined the build process
- Polished the product
- Built for deployment
- Packaged the product

Now . . .

- Ship it!

For More Information

- Apple Developer Connection tools page
  http://developer.apple.com/tools/
- Project Builder page
  http://developer.apple.com/tools/projectbuilder/
- Apple Developer Connection downloads
  http://connect.apple.com/
- Bug Reporting
  http://developer.apple.com/bugreporter/

Roadmap

FF015 Development Tools: Make your thoughts known	Room J1 **Fri., 3:30pm**
909 Debugging in Mac OS X: Learn about gdb and debugging techniques	Hall 2 **Fri., 5:00pm**

Who to Contact

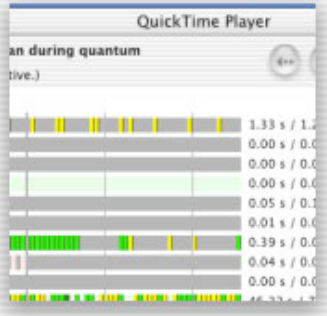
Godfrey DiGiorgi
Technology Manager, Development Tools
ramarren@apple.com

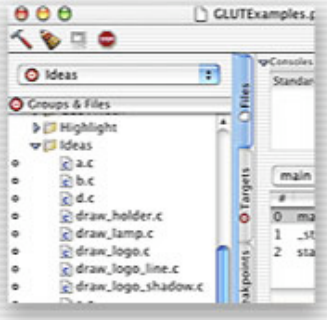
Mailing Lists at Apple
http://lists.apple.com
projectbuilder-users

Development Tools Engineering Feedback
macosx-tools-feedback@group.apple.com

http://developer.apple.com/wwdc2002/urls.html

Q&A





Godfrey DiGiorgi
Technology Manager, Development Tools
ramarren@apple.com

http://developer.apple.com/wwdc2002/urls.html


WWDC2002


WWDC2002


WWDC2002